

## **REMARKS/ARGUMENTS**

Claims 1-7 and 9-20 are pending in the present application. Claims 1, 3, 10-12 and 17-19 have been amended, and Claim 8 has been cancelled, herewith. Reconsideration of the claims is respectfully requested.

### **I. Objection to Claims**

Claims 3 and 19 were objected to as containing typographical errors. Applicants have amended such claims to correct these typographical errors.

Therefore, the objection to the claims has been overcome.

### **II. 35 U.S.C. § 101**

Claims 11-20 stand rejected under 35 U.S.C. § 101 as being directed towards non-statutory subject matter. This rejection is respectfully traversed.

With respect to Claim 11 (and dependent Claims 12-16), the Examiner notes concern that such claim might be interpretable to be a computer listing per se, which is not a physical thing. Applicants have amended Claim 11 to explicitly recite a computer interface, which is a physical thing as it receives user input. Therefore, Claim 11 is not directed to a computer listing per se.

With respect to Claim 17 (and dependent Claims 18-20), the Examiner notes that such claims may be (i) interpretable to be a computer listing per se, and (ii) the computer readable medium may be an electromagnetic signal. Applicants urge error in such assertions, as will now be shown in detail.

Claim 17 recites that the computer program product is encoded in a computer readable medium and operable in a data processing system for testing a software program, as specifically allowed for per the requirements of MPEP 706.03(a) and 2106. See, in particular, MPEP 2106(IV)(B)(1)(a) where it states:

“A claimed computer-readable medium encoded with a data structure defines structural and functional interrelationships between the data structure and the computer software and hardware components which permit the data structure's functionality to be realized, and is thus statutory. See *Lowry*, 32 F.3d at 1583-84, 32 USPQ2d at 1035.”

Accordingly, as Claim 17 expressly recites a computer program product encoded in a computer readable medium and operable in a data processing system for testing a software program, it is shown that

Claim 17 (and similarly for Claims 18-21) is directed to statutory subject matter, pursuant to the USPTO's own MPEP rules.

Still further, Claim 17 explicitly recites a computer program product encoded in a computer readable medium and operable in a data processing system for testing a software program, which is either a 'manufacture' or a 'composition of matter', both of which are statutorily recognized subject matter<sup>1</sup>. In addition, since Claim 17 explicitly recites a computer program product encoded in a computer readable medium and operable in a data processing system, such claim does *not* fall within one of the three judicially determined exceptions of: natural phenomenon, law of nature or abstract idea (see, e.g., MPEP 2106 and in particular MPEP 2106(IV)(B) and (C)), but instead is limited to a practical application in the technological arts<sup>2</sup>. Thus, it is further shown that Claim 17 has been erroneously rejected under 35 U.S.C. § 101 as the invention recited therein does not fall within a judicial exception but instead is limited to a practical application in the technological arts.

The Examiner notes concern that the claimed computer-readable medium may encompass transmission-type media, which the Examiner asserts to be non-statutory since it is alleged to not fall within a statutory category of invention. Appellants respectfully submit that both *In re Lowry*, *Id.* and the MPEP explicitly state that computer-readable medium encoded with a data structure is statutory – and such statement is unconditional *without any type of transmission-media exception* as now alleged by the Examiner to be the current state of the law. Because it is permissible to claim information embodied in a storage medium (*In re Beauregard*, 53 F.3d 1583 (Fed. Cir. 1995)), it is worth noting that the "difference between information storage and information communication is fundamentally only a difference in one's inertial frame of reference." Michael P. Frank, "The Physical Limits of Computing," *Computing in Science & Engineering*, May/June 2002, at 24. The following six cases conclusively establish judicial precedent that electrical signals – such as transmission-type media - are physical, and statutory under 35 U.S.C. § 101.

In *AT & T Corp. v. Excel Communications Inc.*, 172 F.3d 1352, 50 USPQ2d 1447 (Fed. Cir 1999), the CAFC stated at one point about electrical signals being physical:

---

<sup>1</sup> **35 U.S.C. 101 Inventions patentable.**

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

<sup>2</sup> *Only when* the claim is devoid of any limitation to a practical application in the technological arts should it be rejected under 35 U.S.C. § 101. Compare *Musgrave*, 431 F.2d at 893, 167 USPQ at 289; *In re Foster*, 438 F.2d 1011, 1013, 169 USPQ 99, 101 (CCPA 1971).

The Arrhythmia court reasoned that the method claims qualified as statutory subject matter by noting that the steps **transformed physical, electrical signals from one form into another.**

Turning to Arrhythmia Research Technology Inc. v. Corazonix Corp., 958 F.2d 1053, 22 USPQ2d 1033 (Fed. Cir. 1992), the CAFC wrote about electrical signals being physical:

These claimed steps of "converting", "applying", "determining", and "comparing" are physical process steps that transform one physical, electrical signal into another. **The view that "there is nothing necessarily physical about 'signals' is incorrect, citing *In re Taner*, 681 F.2d 787 (CAFC 1982) (emphasis added by Appellants).**

Turning to *In re Taner*, Id., where the PTO was fighting an appeal of a rejection of the PTO Board of Appeals of a claim for a signal, the CCPA (the predecessor court to the CAFC) wrote:

Though the [PTO] board conceded that appellants' process includes conversion of seismic signals into a different form, it took the position that "there is nothing necessarily physical about 'signals'" and that "the end product of [appellants' invention] is a mathematical result in the form of a pure number." That characterization is contrary to the views expressed by this court in *In re Sherwood*, 613 F.2d 809 (CCPA 1980) and *In re Johnson*, 589 F.2d 1020 (CCPA 1978), **where signals were viewed as physical and the processes were viewed as transforming them to a different state.** ... and in *Sherwood* expressly recognized that **"seismic traces are ... physical apparitions."** 613 F.2d at 819. That those "physical apparitions" may be expressed in mathematical terms is in our view irrelevant (emphasis added by Appellants).

The last case is the Supreme Court decision *O'Reilly v. Morse* from 1853 (56 U.S. 62), in which the Supreme Court upheld the following product claim for signals:

1. I claim as my invention the system of signs consisting of dots spaces and of dots, spaces and horizontal lines for numerals, letters, words or sentences substantially as herein set forth and illustrated for telegraph purposes.

So, across decades of judicial decisions, we have the CAFC and the Federal Circuit repeatedly stating that electrical signals are physical, backed up by the Supreme Court. Being physical, such signals are tangible articles. Since such signals can be manufactured according to numerous varieties of technological methods, such signals are articles of manufacture or composition of matter, both of which are statutory categories of patentability under 35 U.S.C. § 101. Thus, Claim 17 is shown to be statutory under 35 U.S.C. § 101 as it explicitly recites a computer program product encoded in a computer readable medium and operable in a data processing system for testing a software program, pursuant to both (extensive) judicial case law and the USPTO's own MPEP rules. Accordingly, Claim 17 (and dependent Claims 18-21) has been erroneously rejected under 35 U.S.C. § 101.

Therefore, the rejection of Claims 11-20 under 35 U.S.C. § 101 has been overcome.

### **III. 35 U.S.C. § 102, Anticipation**

Claims 1-6, 8-12, 16-18 and 20 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Rojas et al. (US Patent No.: 6,425,123). This rejection is respectfully traversed.

With respect to Claim 1, Applicants have amended such claim to include the features previously recited in Claim 8 (which is thus being cancelled herewith, without prejudice or disclaimer). As amended, Claim 1 recites “responsive to a user input containing text in a source language, entered by a user through a computer interface in the data processing system, translating the text from the source language to the target language to form translated text” and “automatically inserting the translated text into a user interface of the software program to be tested to form inserted, translated text, wherein the software program is written in the target language, and *wherein the inserting step is initiated in response to another user input to commit the translated text displayed on the computer interface*”. As can be seen, Claim 1 is directed to specific user input co-action for testing a software program, where (1) text to be translated is input and automatically translated to form translated text, and (2) this translated text is then inserted into a user interface of the software program to be tested, *with the translated text insertion being done in response to another user input to commit the translated text*. This two-pronged user-initiated action thus advantageously provides an ability for the user to provide an intermediate confirmation that the translated text, translated from the user input provided text, is in correct form to be sent to the software program under test *prior* to its being sent to the software program under test. In contrast, per the teachings of the cited reference, no such intermediary user confirmation is provided. Instead, base text to

be translated is placed in a localization file (Rojas col. 4, lines 53-56). Then, a mock translation process is executed on this localization file (Rojas col. 4, lines 57-58). This localized file containing the mock translation is then used by the software application under test (col. 5, lines 34-37). The user is then able to view a resulting screen shot generated by the application program under test to determine if the application program properly handles this translation (col. 5, lines 38-43). Importantly, the only user interaction with any type of translated text is only at the very end of the process, where the application program under test generates a screen output which the user verifies. In contrast, per the features of Claim 1, an intermediary stage is provided in real-time such that the user can dynamically verify proper translation *prior* to the translated text being provided to the software program under test. This can be particularly useful where the translation is a mock translation and there can be different variations in how the text is actually translated. While the cited reference is similarly directed to mock translations, *it does not provide this additional fail-safe or double-check confirmation of proper translation prior to its use by the software program under test.* Instead, the only user involvement with the translated text is after it has already been processed by the software application being tested, as described by Rojas at col. 6, lines 20-30:

“FIG. 4 shows a sample application display screen 400. Note that this screen is entirely in standard English, including each of the “buttons” at the bottom of the screen, e.g., button 410, and including menu options 420.

Referring now to FIG. 5 (and with reference also to FIG. 4), since the mock translation has distinct beginning and end characters, **it becomes a simple process for the user or programmer to check each screen of the executing application** to determine if any characters are missing from the beginning or end of the “mock-translated” text.”  
(emphasis added by Applicants)

Thus, it is urged that amended Claim 1 is not anticipated by the cited reference as all claimed features are not taught by the cited Rojas reference, such as the claimed feature of “automatically inserting the translated text into a user interface of the software program to be tested to form inserted, translated text, wherein the software program is written in the target language, and *wherein the inserting step is initiated in response to another user input to commit the translated text input*”.

Applicants initially traverse the rejection of Claims 2-6 and 9 for reasons given above with respect to Claim 1 (of which Claims 2-6 and 9 depend upon).

Further with respect to Claim 2, such claim recites “wherein the computer interface includes an option for a user to enable and disable translation of the text”. It is urged that the cited reference does not teach a computer interface that includes an option for a user to enable/disable translation of the text. In rejecting Claim 2, the Examiner cites Rojas Figures 4 and 5 and related text as teaching this claimed feature. Applicants urge that the related text for Figures 4 and 5, found at col. 6, line 20 – col. 7, line 46, does not describe any such option. Instead, Rojas states:

“FIG. 4 shows a sample application display screen 400. Note that this screen is entirely in standard English, including each of the “buttons” at the bottom of the screen, e.g., button 410, and including menu options 420.

Referring now to FIG. 5 (and with reference also to FIG. 4), since the mock translation has distinct beginning and end characters, it becomes a simple process for the user or programmer to check each screen of the executing application to determine if any characters are missing from the beginning or end of the “mock-translated” text.

Furthermore, since the mock-translated text has been expanded, using placeholder characters, to meet internationalization guidelines, it is also now a simple matter to examine each screen for alignment errors or other formatting errors.

Any hard-coded text, which has not been put through the mock-translation process, will also be apparent since there will be no beginning or end markers or placeholder characters. Note, for example, the “Add With Defaults” button. In FIG. 4, of course, this button 410 is all plain English text. In mock-translated FIG. 5, however, it is clear that corresponding button 510 has been mock translated, since brackets and placeholder characters are visible. Menu items 420 in FIG. 4 are similarly mock-translated as menu items 520 in FIG. 5. Note, conversely, that the “Universal” menu item 530 appears exactly as in FIG. 4 as menu item 430; this text has therefore been hard-coded, and this error can be easily spotted and repaired.

Another common error, not shown here, which may be easily detected using this mock-translation technique, is the presence of labels or other text which is composed of two or more separately-translated text strings. Because many foreign languages, when translated from English, will rearrange the word order of subject, objects, and verbs, each phrase to

be translated should be translated as a whole if it is to be displayed correctly in other languages. For this reason, composed text must be eliminated. Using the mock-translation techniques described herein, it is a simple matter for the software programmer or developer to spot composed text, since placeholder characters will appear between words of a single string of mock-translated text. If brackets or other indicators are used to denote the beginning and/or end of the mock-translated text, the appearance of these indicators with each separate piece of text will indicate text composed of piecemeal parts.

Note that in FIGS. 4 and 5, the tilde placeholder has been replaced with a double-wide dash (--). This illustrates another innovative mock-translation technique, useful when the software is to be translated into Japanese or other languages that use double-byte character sets.

The United States and other countries which use a standard ASCII character set require only a single byte to identify individual characters. Some other languages, because they are more extensive than English, use a double-byte character set for language generation. Translation of single-byte languages into a double-byte character set for foreign use involves additional concerns because it is possible that the double-byte character may be read as two single-byte characters.

One specific (and notorious) example is the "5C" problem; many double-byte characters have "5C" as the second byte, but "5C" represents a backslash character (.backslash.) in a single-byte character set. Therefore, many double-byte characters may be incorrectly displayed as a different character followed by a backslash.

The mock translation system provides a solution to this problem, by performing a mock translation as described above, but using double-byte characters for the brackets and placeholder characters. By using a double-byte, double-wide dash character (character 815C) as the placeholder character, double-byte translation problems will also be evident on visual inspection. The double-wide dash character itself is subject to the "5C" problem, so if the display of double-byte characters is problematic, backslash characters will be visible in the placeholder character field. Note that in FIG. 5, translated menu items 520 appear correctly with placeholder dashes, and no backslash characters are visible; this indicates that the mock-translation (for these items) was performed correctly.

Further, in this embodiment, the double-byte, double-wide open and close brackets can be used as field boundaries.

This process provides the advantages of the basic mock-translation system, with additional capabilities for detecting double-byte problems. Because, after mock translation, the double-byte characters will be present in the visual display of the program, any errors in the program relating to double-byte characters are immediately apparent, instead of much further down the development process, when translation of the program is usually performed. Again, the localization files remain readable to English-speakers, and now allow the software developer to easily check for internationalization problems.”

As can be seen, these cited passages do not describe any type of computer interface that includes an option for a user to enable and disable translation of the text. Instead, the computer interface that is described is the actual translation results themselves (Figure 5) which resulted from translating the base-language screen (as depicted in Figure 4). A review of the Figure 4 screen shows an ability to add a ‘monitor’ to a distributed monitoring profile. There are user options for obtaining more information ‘About This Collection’ and ‘About This Monitor’. There are also user options for adding the monitor ‘With Defaults’ or ‘Empty’ as well as user options to ‘Cancel’ the action and to receive ‘Help’ (these options depicted at the bottom of Figure 4, at element 410). There is also a field for a user to select ‘Monitoring Collections’ (element 430), ‘Monitoring Sources’ (element 420), and ‘Monitor Arguments’ and an associated ‘Attributes’ field for such Argument monitoring. Importantly, none of the user selections pertain to any option that a user can select to enable and disable translation of the text. That is because this screen is not any type of translation tool control screen but instead is the interface screen for *actual application program that is being tested*. The features of Claim 2 are directed to a computer interface for a tool that is used to dynamically accept and translate text that is *subsequently sent* to an application under test. While such application under test may be similar to what is depicted in Figure 4, such an application under test is very different from a test/control front-end tool that is used to generate translated text that is then subsequently sent to such a back-end application program being tested.

Nor does the Figure 5 screen overcome this teaching deficiency, as Figure 5 has the same user fields as described above with respect to Figure 4, albeit being displayed in a mock translation form. Quite simply, the fact that one screen (Figure 5) is a translation of another screen (Figure 4) does not teach a computer interface that includes an option for a user to enable and disable translation of the text,



as expressly recited in Claim 2. Thus, it is further urged that Claim 2 has been erroneously rejected as there are additional claimed features that are not taught by the cited reference.

Applicants traverse the rejection of Claims 10 and 11 (and dependent Claims 12 and 16) for similar reasons to those given above with respect to Claim 1.

Further with respect to Claim 12 (and dependent Claim 16), such claim recites “wherein the computer interface includes options for a user to select both a source language and a translation mode at any time”. The cited reference does not teach any type of computer interface that includes options for a user to select either a source language or a translation mode at any time. In rejecting Claim 12, the Examiner states that this computer interface feature is taught by Rojas at col. 5, lines 20-22 (since the reasoning given in rejecting Claim 5 is the sole reasoning given in rejecting Claim 12). Applicants urge that at col. 5, lines 20-22, Rojas states (the entire paragraph has been reproduced herewith to provide proper context, with the particular Examiner-cited passage being bolded):

“This allocation of additional characters accommodates the greatest number of extra characters needed for given ranges, according to the IBM internationalization guidelines. This provides a testing method which will be effective for the widest range of potential language translations. **Of course, in practice, those of skill in the art may vary these figures to fit the particular translations that will be made on the software.**”

As can be seen, this passage describes additional character allocation in accordance with particular guidelines, where in some instances 20 characters are added and in other instances more than 20 characters are added (per the table immediately preceding this cited passage). The specific cited passage used in rejecting Claim 12 merely states that these ‘number of additional character’ figures that are depicted in the table are not fixed, rigid numbers but can be varied to fit a particular translation that will be made on the software. *This cited passage does not describe any type of computer interface at all*, and thus this cited passage does not teach the particular computer interface provided by the features of Claim 12 – a computer interface that includes options for a user to select both a source language and a translation mode at any time. Thus, as every element recited in Claim 12 is not identically shown in the Rojas cited reference, it is further shown that Claim 12 (and dependent Claim 16) has been erroneously rejected as being anticipated by Rojas<sup>3</sup>.

---

<sup>3</sup> For a prior art reference to anticipate in terms of 35 U.S.C. 102, every element of the claimed invention must be identically shown in a single reference. *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990) (emphasis added by Applicants).

Applicants traverse the rejection of Claim 17 (and dependent Claims 18 and 20) for similar reasons to those given above with respect to Claim 1.

Applicants further traverse the rejection of Claim 18 for similar reasons to the further reasons given above with respect to Claim 12.

Applicants further traverse the rejection of Claim 20 for similar reasons to the further reasons given above with respect to Claim 2.

Therefore, the rejection of Claims 1-6, 8-12, 16-18 and 20 under 35 U.S.C. § 102(b) has been overcome.

#### **IV. 35 U.S.C. § 103, Obviousness**

Claims 7, 13-15 and 19 stand rejected under 35 U.S.C. § 103 as being unpatentable over Rojas et al. (US Patent No.: 6,425,123) in view of See et al. (US Patent No.: 5,572,668). This rejection is respectfully traversed.

With respect to Claim 7 (and similarly for Claim 15), Applicants traverse the rejection of such claim for similar reasons to those given above with respect to Claim 1 (of which Claim 7 depends upon), and urge that the newly cited See reference does not overcome the teaching deficiencies identified above with respect to Claim 1.

With respect to Claim 13 (and dependent Claim 14), Applicants initially traverse the rejection of such claim for similar reasons to those given above with respect to Claim 12 (of which Claim 13 depends upon), and urge that the newly cited See reference does not overcome the teaching deficiencies identified above with respect to Claim 12.

Further with respect to Claim 13 (and dependent Claim 14), such claim recites “wherein the translation mode includes a default translation and a look up translation”. As can be seen, Claim 13 is directed to particular features/aspects pertaining to the claimed translation mode that is recited in Claim 12, and in particular the two features/aspects of such translation mode are (i) a default translation and (ii) a look up translation. Both of these features/aspects (default translation and look up translation) are with respect to the claimed translation mode that is recited in Claim 12 (i.e. this translation mode is a part of the computer interface that includes an option for a user to select such a translation mode). In rejecting Claim 13, the Examiner cites Rojas col. 2, lines 48-62 and See col. 2, lines 19-23 as teaching all of the features of Claim 13. Applicants urge that the cited Rojas passage at col. 2 is a general summary of the Rojas invention, and *does not describe any type of computer interface* that includes an option for a user to select anything, and so this cited passage does not describe any type computer interface that includes an option for a user to select a translation mode that includes either a default translation or a look-up

---

translation (and per the features of Claim 13, both are required). Nor does the cited See passage overcome such teaching deficiency. There, See describes *an internal lookup table* that is used to map ASCII characters to problem characters. This cited See passage *does not describe any type of computer interface* that includes an option for a user to select anything, and so this cited passage does not describe any type computer interface that includes an option for a user to select a translation mode that includes either (i) a default translation or (ii) a look-up translation (and per the features of Claim 13, both are required). Thus, the Examiner has failed to establish a prima facie showing of obviousness with respect to Claim 13, and accordingly the burden has not shifted to Applicants to rebut such improper obviousness assertion<sup>4</sup>. In addition, as a proper prima facie showing of obviousness has not been established, Claim 13 (and dependent Claim 14) has been erroneously rejected<sup>5</sup>.

Applicants traverse the rejection of Claim 19 for similar reasons to those given above with respect to Claim 13.

Therefore, the rejection of Claims 7, 13-15 and 19 under 35 U.S.C. § 103 has been overcome.

---

<sup>4</sup> To establish prima facie obviousness of a claimed invention, all of the claim limitations must be taught or suggested by the prior art. MPEP 2143.03. *See also, In re Royka*, 490 F.2d 580 (C.C.P.A. 1974). In rejecting claims under 35 U.S.C. Section 103, the examiner bears the initial burden of presenting a prima facie case of obviousness. *In re Oetiker*, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992). Only if that burden is met, does the burden of coming forward with evidence or argument shift to the applicant. *Id.*

<sup>5</sup> If the examiner fails to establish a prima facie case, the rejection is improper and will be overturned. *In re Fine*, 837 F.2d 1071, 1074, 5 USPQ2d 1596, 1598 (Fed. Cir. 1988).

V. **Conclusion**

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: May 23, 2007

Respectfully submitted,

/Wayne P. Bailey/

Wayne P. Bailey  
Reg. No. 34,289  
Yee & Associates, P.C.  
P.O. Box 802333  
Dallas, TX 75380  
(972) 385-8777  
Attorney for Applicants